Amendments to the Claims:

1. (Currently Amended) A method for managing type information for operands, the method comprising:

accomplishing the following results through execution of a single register instruction in a register of a processor:

adding an operand tag to a tag stack; and

updating a stack pointer for the tag stack to recognize the addition of the operand tag to the tag stack.

2. (Currently Amended) A method according to claim 1, wherein the single register instruction comprises a shift instruction.

3. (Original) A method according to claim 2, wherein the shift instruction comprises a rotate instruction.

4. (Currently Amended) A method according to claim 1, further comprising:

accomplishing the following results through execution of one register instruction:

removing an operand tag from the tag stack; and

updating the stack pointer for the tag stack to recognize the removal of the operand tag from the tag stack.

5. (Currently Amended) A method according to claim 4, wherein the one register instruction comprises a shift right instruction.

6. (Original) A method for managing type information for operands, the method comprising:

shifting a bit value of 1 into a register, in conjunction with creation of a reference operand; and

shifting a bit value of 0 into the register, in conjunction with creation of a non-reference operand.

7. (Currently Amended) A method according to claim 6, wherein:

the register serves as a tag stack register, the tag stack register to be used for storing a stack of operand tags, each operand tag to indicate whether a corresponding operand on an operand stack is to be treated as a reference operand or a non-reference operand; and

the method further comprises initializing the tag stack register by:

assigning a low order bit of the tag stack register to a value of 0; and

assigning ~~substantially all~~ other bits of the tag stack register to a value of 1.

8. (Original) A method according to claim 6, further comprising:

using shift left operations to shift bit values into a low order bit of the register in response to operands being added to an operand stack.

9. (Currently Amended) A method according to claim 6, further comprising:

right shifting bit values in the register in conjunction with removal of an [[the]] operand, the operand being one of the reference operand and the non-reference operand.

10. (Original) A method according to claim 9, further comprising:

shifting the bit value of 1 into a high order bit of the register in conjunction with removal of the operand.

11. (Original) A method according to claim 6, wherein:

the register serves as a tag stack register, the tag stack register to be used for storing a stack of operand tags, each operand tag to indicate whether a corresponding operand on an operand stack is to be treated as a reference operand or a non-reference operand; and

the method further comprises:

treating a highest order bit with the value of 0 in the tag stack register as a stack pointer; and

determining a depth of the stack of operand tags, based at least in part on a location of the stack pointer.

12. (Currently Amended) A processing system with control logic for managing type information for operands, the processing system comprising:

a processor;

a machine-accessible storage medium responsive to the processor; and

instructions in the machine-accessible storage medium, the instructions to implement at least part of a virtual machine when executed by a processing system, the virtual machine to accomplishing the following results through execution of a single register instruction:

adding an operand tag to a tag stack; and

updating a stack pointer for the tag stack to recognize the addition of the operand tag to the tag stack.

13. (Currently Amended) A processing system according to claim 12, wherein the single register instruction to be used by the virtual machine to add the operand tag to the tag stack and to update the stack pointer comprises a shift instruction.

14. (Original) A processing system according to claim 13, wherein the shift instruction comprises a rotate instruction.

15. (Currently Amended) A processing system according to claim 12, the virtual machine further to accomplish the following results through execution of one register instruction:

removing an operand tag from the tag stack; and

updating the stack pointer for the tag stack to recognize the removal of the operand tag from the tag stack.

16. (Original) A processing system according to claim 12 wherein the processor supports a little-endian byte order.

17. (Currently Amended) An apparatus containing control logic for managing type information for operands, the apparatus comprising:

a machine-accessible storage medium; and

instructions in the machine-accessible storage medium, the instructions to implement at least part of a virtual machine when executed by a processing system, the virtual machine to accomplishing the following results through execution of a single register instruction:

adding an operand tag to a tag stack; and

updating a stack pointer for the tag stack to recognize the addition of the operand tag to the tag stack.

18. (Currently Amended) An apparatus according to claim 17, wherein the single register instruction to be used by the virtual machine to add the operand tag to the tag stack and to update the stack pointer comprises a shift instruction.

19. (Original) An apparatus according to claim 18, wherein the shift instruction comprises a rotate instruction.

20. (Currently Amended) An apparatus according to claim 17, the virtual machine further to accomplish the following results through execution of one register instruction:

removing an operand tag from the tag stack; and

updating the stack pointer for the tag stack to recognize the removal of the operand tag from the tag stack.